

REVISTA INCAING

ISSN 2448 9131



Análisis de eficiencia de la red neuronal convolucional (cnn) y el sistema de aprendizaje tensorflow

Fernando Alberto Arévalo Mesa A¹, Juan David Ojeda Bernal A², Paula Camila Enríquez Galindo A³, Cesar Yesid Barahona Rodríguez A⁴

Universidad de Cundinamarca Extensión Facatativá

1,2,3: Estudiantes de la Universidad de Cundinamarca

4: Profesor de la Universidad de Cundinamarca y director del proyecto

Resumen. La tecnología de la identificación de imágenes y aprendizaje automático se ha aplicado en varios campos como medicina, en las plantas, reconocimiento de especies (serpientes)[1][2]. La aplicación de identificación de imágenes actualmente en la detección e identificación de objetos en determinar una imagen de forma momentánea. En este artículo se propone evaluar el comportamiento de las redes neuronales convolucionales junto a tensorflow, con el objetivo de ver cómo se comporta el sistema y si este es eficiente o no ante cada una de las pruebas que se le realizó al sistema con las diferentes imágenes sé que pasaron, con el fin de ser si es sistema muestra números de eficiencia o no. De esta manera veremos los resultados obtenidos durante cada una de las pruebas que se realizaron, de esta forma llegaremos a la conclusión si el sistema es eficiente o no. Después de que el sistema ha captado imágenes de diferentes especies, de esta manera mejora la capacidad de identificación del sistema de una manera ágil. Implementando estas tecnologías (red neuronal convolucional (CNN) y

el sistema de aprendizaje tensorflow), realizando la unión de estos dos se obtuvo un sistema más rápido en los resultados del reconocimiento de especies.

La presente investigación se ha dedicado al estudio del comportamiento de la red neuronal y el sistema de aprendizaje tensorflow, del cual se concluye con evidencias positivas el funcionamiento del sistema durante el periodo de pruebas. Se encontraron fallos mínimos principalmente errores humanos al momento de tomar los registros fotográficos como el desenfoque, la baja calidad o el plano en el que se encuentra el animal en la imagen que se desea cargar, en el caso de las imágenes de alta resolución y calidad no se encontraron fallos a gran escala. Dando a entender que la red neuronal y el sistema de aprendizaje presentan un alto porcentaje de asertividad reflejado en las pruebas realizadas.

Palabras Clave: Red neuronal convolucional (CNN), sistema de aprendizaje tensorflow, reconocimiento, especies caninas

I. Introducción

Este artículo consiste en una serie de pruebas de funcionamiento del reconocimiento de imágenes a partir de una red neuronal junto al sistema de aprendizaje tensorflow. En primera instancia se inicia con la fase de pruebas, a partir de la carga de imágenes de prueba al aplicativo que consume la red neuronal convolucional.

El reconocimiento de patrones a partir de imágenes abarca grandes áreas de investigación, dentro de las cuales se encuentran los sistemas basados en redes neuronales que se emplean para mejorar el reconocimiento de especies dentro del aplicativo. Por consiguiente, en este documento se propone un sistema inteligente de reconocimiento de especies caninas donde el usuario solo con ingresar una fotografía del animal permitirá que el sistema realice el reconocimiento y clasificación según los rasgos de diferentes razas con los cuales se lograron identificar la mayor cantidad de similitudes. Para lograr esto el algoritmo de la red neuronal convolucional (CNN) detecta los patrones del animal en la imagen y por medio de la implementación de Tensorflow como sistema de aprendizaje, dará la clasificación más cercana de la especie.

Tensorflow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, desarrollada por Google para construir y entrenar redes neuronales y descifrar patrones, esto empieza con la importancia de las librerías de alto rendimiento para el análisis numérico como son Eigen (una librería de alto rendimiento para C++ y CUDA) y CuDNN (una librería de Nvidia para redes neuronales profundas) [3] esto facilita las operaciones con matrices y vectores.

Además, CNN es un tipo de modelo de aprendizaje que procesa datos que tienen un patrón de cuadrícula, como lo son las imágenes y este es creado para que aprenda de forma automática, ya

que adapta las jerarquías desde patrones de bajo a alto nivel. La arquitectura del sistema de la red neuronal (CNN) es la tecnología de identificar imágenes por medio de su arquitectura de red

Faster-RCNN (busca las regiones de interés en la imagen de forma ágil), el algoritmo maneja tres capas de operaciones: red de características, red de propuesta de región (RPN) y red de detección. Una de las capas se utiliza para la clasificación, las otras dos se usan para encontrar las regiones que contienen los píxeles característicos de la imagen, podemos ver en la de como trabaja la red neuronal

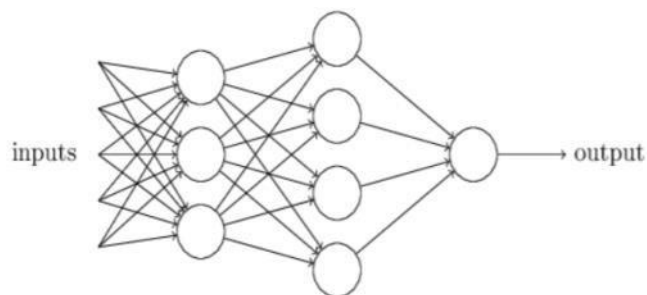


Figura 1. Estructura de una red neuronal simple. [4]

En la figura 1 hay que ver las capas de una red neuronal simple, cada círculo representa una red neuronal, esta estructura consta de tres capas que son: una capa de entrada, capas ocultas y una capa de salida. Una capa de entrada recibe la información, las capas ocultas hacen el proceso y las capas de salida obtienen el resultado final. Las capas se comunican entre señales, así sean capas ocultas o capas de entrada y salida.

Se realizaron pruebas a la librería y a la red neuronal convolucional, para demostrar el funcionamiento del algoritmo, al analizar 120 razas se demostró que de 26.936 imágenes el 78% de las imágenes analizadas fueron reconocidas de una

manera satisfactorio y el 22% no fueron reconocidas de manera satisfactorio.

Para llegar a esta información estadística se llevo a cabo un análisis extenso a los resultados recolectados en el sistema estadístico en el cual se recogieron los datos y se clasificaron.

prueba que se utilizaron fueron sacadas de la pagina Stanfor Dogs Dataset, donde se encontraron 26.936 imágenes,120 razas de perros analizadas las cuales fueron utilizadas como objeto de prueba.

La recolección de los datos se hizo por medio de un sistema estadístico, en el cual se realizaron mapas de calor con los datos obtenidos, de esta forma se clasifican los datos en deficiente, insuficiente, buena, sobresaliente. Por medio de esta clasificación de datos se pudo determinar si el sistema es eficiente o de lo contrario es deficiente. Estos datos pasaron por un análisis estadístico, donde se clasificaron los datos de cada una de las razas de perros que se analizaron.

Además, se logró llegar a las conclusiones de porque el sistema no puede realizar el reconocimiento de una imagen, cuáles pueden ser las posibles causas y como se puede solucionar cada una de las causas que se evidenciaron.

A. Red Neuronal convolucional (CNN)

Ahora el CNN es una construcción de código matemático, que tiene tres tipos de capas o conocidas como bloques de construcción que son: convolución, agrupación y capas completamente conectadas.

Las dos primeras capas anteriormente nombradas (convolución y agrupación), extraen las características de la imagen, estas características lo que realizan es una disminución de datos menos redundante, filtrando así las características de la imagen que genera un mapa con las características, después la última capa extrae las características finales.

$$x^1 \longrightarrow \boxed{w^1} \longrightarrow x^2 \longrightarrow \dots \longrightarrow x^{L-1}$$

Figura 2. Estructura de la CNN [5]

En la figura 2 podemos ver el funcionamiento de las capas de la CNN, x^1 es la entrada de la imagen y w^1 realiza todo el proceso de las dos primeras capas de la CNN, x^2 es la salida con los resultados

II. Materiales y métodos

Para realizar el proceso de pruebas, utilizamos la red neuronal (CNN) junto a tensorflow el cual es un sistema de aprendizaje automatico, las imágenes de

de las dos primeras capas, en el espacio revisa si hay errores en los resultados obtenidos.

L es un vector dimensional, trabaja con x realizando una masa de probabilidad de procesamiento esto se convertiría en x^{L-1} donde esta sería la capa que transforma los datos. w^{L-1} en este punto se realiza toda la parte matemática donde envía a la capa final el análisis correspondiente de los datos, ya convertidos en un mapa sin datos repetidos.

Al proceso realizado anteriormente se le llama kernel, este proceso se hace de una manera repetitiva, por medio de este proceso repetitivo el sistema aprende de manera automática a través de un algoritmo de optimización llamado “retro programación y descenso de gradiente”. De esta forma llega al final del proceso donde z recibe todos los datos o es llamada la capa de salida, esta es una capa oculta dentro de todo el sistema.

Teniendo claro cómo funciona la CNN, vamos a ver como hace el reconocimiento de imágenes por medio de código, como es el paso a paso debajo de este a continuación.

La arquitectura del sistema de la red neuronal (CNN) es la tecnología de identificar imágenes por medio de la arquitectura red Faster-RCNN, ya que esta maneja tres capas de operaciones las cuales son: red de características, red de propuesta de región (RPN) y red de detección

Este sistema extrae mapas de la imagen a través de capas convolucionales. Después, la red propuesta de región (RPN) procesa los mapas y proporciona información a ROI (Regiones que pueden contener puntos de características). De esta manera se permite a ROIhead (responsable de procesar las respuestas de ROI y las propuestas de RPN), en este punto se revisa la información que tiene ROI y se hace la corrección de coordenadas. Ya para terminar la red de detección toma las entradas de esta y RPN genera la clasificación de la mascota.

De esta manera lograremos que el sistema realice el reconocimiento de especies en tiempo real generando un entrenamiento en las capas de detección de imágenes y RPN que realiza el reconocimiento de estas. Identificando el RPN sería nuestra z en la figura 2, la cual obtiene el resultado final de todo el proceso que realiza la red neuronal entre las capas de la misma.

cuando ya se ha entrenado la red neuronal, cuando se ha realizado la red neuronal pasa a ser red neuronal Faster-RCNN.

R-CNN (red neuronal convolucional basada en regiones), empieza a identificar de manera más rápida los puntos de análisis de las imágenes, RCNN empieza a identificar de manera rápida y efectiva los puntos, las partes claves de las mascotas, también realiza el mapeo de las imágenes.

B. Explicación matemática de las redes neuronales convolucionales (CNN)

La convolución se define como el producto de la integral de dos funciones desplazando una función t (t es un tiempo continuo), y se denota de la siguiente manera:

$$F * g \int_{-\infty}^{\infty} f(T)g(t - T)dT$$

Figura 3. Explicación matemática de la convolución de las redes neuronales convolucionales [6]

Como se puede ver en la figura 3, hay una integral la cual se resuelve con las mismas reglas de la integración, haciendo esa solución llegamos a la siguiente ecuación:

$$F * (f * g) = (F(f)) * (F(g))$$

Figura 4. Teorema de convolución [6]

Como se puede observar en la figura 3 podemos ver que el resultado de la integral de la figura 4 es el teorema de convolución o la más conocida transformada de Fourier de f, este también aplica al teorema de Laplace.

Todo esto desencadena en un producto matricial, como se puede evidenciar en la figura 5, donde podemos observar el operador de convolución aplicado sobre un filtro. Para llegar al resultado se

Además, este sistema utiliza un conjunto de datos COCO (Common Object in Context o su traducción en español que es Objeto común en contexto), esto ayuda al sistema a hacer un reconocimiento de las imágenes más rápido, a esto se le llama red neuronal Faster-RCNN, se logra

tiene que multiplicar cada operador por cada elemento del filtro convolucional así se obtiene el resultado final de la matriz.

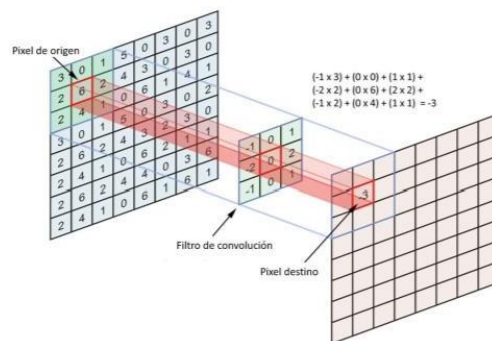


Figura 5. Teorema de convolución [6]

De esta manera se puede evidenciar la estructura matemática de las redes neuronales convolucionales y el resultado final se puede ver en la figura 6, donde vemos cómo funciona de una manera clara la red neuronal convolucional, donde se puede evidenciar en la entrada esta la imagen, mapa de características donde pasa la imagen y se empieza a revisar toda su estructura, se toman los datos y se para a la sección de convoluciones donde se empieza a quitar los números repetidos en las regiones. En la sección de submuestra se realiza el reconocimiento y en la salida nos da como resultado nos da el reconocimiento el sistema.

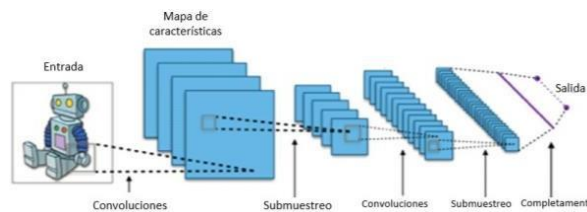


Figura 6. Teorema de la red neuronal convolucional [6]

C. Tensorflow un sistema de aprendizaje automático

TensorFlow una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, desarrollado por Google para construir y entrenar redes neuronales y descifrar patrones y

Nvidia para redes neuronales profundas)[3] esto facilita las operaciones con matrices y vectores.

Esta librería trabaja junto a las redes neuronales capaces de reconocer patrones es el funcionamiento del proyecto de IA, las operaciones que se realizan sobre los arrays de datos o también llamados librerías son una analogía a la estadística inferencial, con una amplia base de datos, que en este caso es la información de cada raza se intenta reconocer el patrón de estas.

En un principio lo más importante es tener los datos, una librería amplia y variada de referencias que guíen la red neuronal hacia una interpretación

Tensorflow utiliza grafos para crear un modelo sobre el que trabaja donde cada nodo representa una operación aritmética que genera los tensores, (elementos que da nombre a esta librería) son objetos geométricos que describen relaciones entre vectores geométricos, escalares y otros tensores. Es decir, son los objetos que maneja la red neuronal para producir valores [7]. Históricamente, sin embargo, los tensores han hecho menos avances en la informática, tradicionalmente se ha asociado más con la matemática discreta y la lógica. Este estado de cosas ha comenzado a cambiar significativamente con la llegada del aprendizaje de máquina. El aprendizaje de máquina moderno se basa en la manipulación y el cálculo de los tensores. El ejemplo más simple de un tensor es un escalar, un único valor constante extraído de los números reales, llamamos a un escalar un tensor de rango 0. Si los escalares son tensores de rango 0, ¿qué es un tensor de rango 1? Un tensor de rango 1 es un vector; Una lista de números reales (a, b). Un tensor de rango 3 se formaría (N, N, N). Un elemento arbitrario del tensor se seleccionaría especificando (i, j, k) como índices[1].

Tensorflow permite la definición de varios gráficos, pero la mayoría de los programas solo usan el valor predeterminado one, que esta disponible en esta librería como `tf.get_default_graph()`. Cuando una operación se

correlaciones, inicia con la importación de las librerías que necesitaremos para construir el programa, Tensorflow cuenta con librerías de alto rendimiento para el análisis numérico como son Eigen (una librería numérica de alto rendimiento para C++ y CUDA) y cuDNN (una librería de

asertiva de los patrones de datos, con este conjunto comenzamos a entrenar la red neuronal, clasificando nuestros datos mediante operadores condicionales o lógica booleana, estos datos discretos serán nuestro conjunto de datos de entrenamiento.

Con el aprendizaje supervisado, el objetivo es que la librería aprenda o identifique un patrón a partir de un conjunto de datos que se utilizan para su entrenamiento, permitiendo realizar predicciones de conjuntos de datos no observados previamente. De esta forma los datos iniciales, se usan como entrada para entrenar la red neuronal convolucional.

crea, se agrega automáticamente al gráfico predeterminado (que está vacío al inicio), si se desea agregar gráficos personalizados, se debe anular el gráfico predeterminado dentro de su alcance.

Para diferenciar los datos de entrenamiento con los de prueba se usa el método de validación cruzada o también conocido como cross-validation. Es una técnica para evaluar los resultados de un análisis estadístico y poder garantizar que son independientes de la partición entre el conjunto de datos que se utiliza para entrenamiento y el conjunto de prueba. Es utilizado en entornos donde el objetivo es la predicción y se quiere estimar cómo de preciso es el modelo generado este da una distribución adecuada de cada clase para el mejor entrenamiento de las mismas.

Las sesiones son la segunda mitad del proceso de TensorFlow estas ejecutan las operaciones que son llamadas por los objetos y deben construirse después de que se hayan agregado todas las operaciones al gráfico y, solo entonces se pueden ejecutar sus operaciones.

El constructor de la sesión toma un argumento opcional que lo enlaza con él con la parte del gráfico que quiere que se ejecute (si no especificamos el gráfico, se utiliza el gráfico predeterminado). Una sesión solo puede ejecutar

operaciones que estén en un gráfico, y este mismo grafico puede ser ejecutado en diferentes sesiones.

Para evitar la pérdida de información TensorFlow ofrece un tipo de programa persistente de nodos que serían parecidas en función a las variables. El constructor de un tf. La variable tf recibe muchos cuando la inicializamos. dtype especifica el tipo de datos que almacenará la variable (y le dice al constructor que convierta la entrada a ese tipo, generando un error si no es posible). Trainable es un parámetro booleano más interesante cuyo valor predeterminado es True, lo que le dice a TensorFlow si la variable debe ser entrenada por optimizadores (por ejemplo, una matriz de peso en una red neuronal).

Una vez que se han definido todas las variables en un gráfico, necesitamos agregar la operación tf.global_variables_initializer (). Esto asigna a cada variable el argumento que se pasó a su constructor, y debería ser la primera operación que ejecutamos una vez que comienza la ejecución. Es posible para asignar un valor particular a una variable durante el tiempo de ejecución también, agregando tf.assign nodos al gráfico. De hecho, cuando llamamos al constructor tf.Variable, TensorFlow agrega tres nodos: la variable en sí, la operación de asignación y el tensor de valor inicial.[8]

argumentos predeterminados (de hecho, no tiene otros que no sean los predeterminados), los más relevantes de los cuales son posiblemente initial_value, dtype y trainable. initial_value recibe cualquier argumento tensor-convertible y lo establece como el valor deseado para esa variable

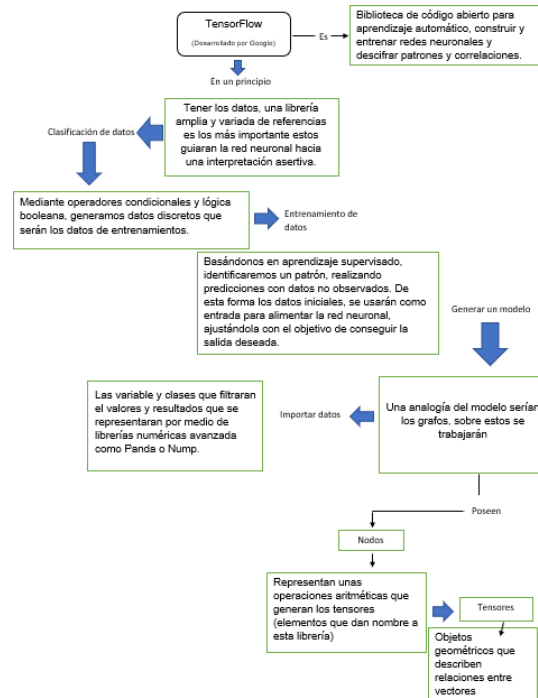


Figura 7. Funcionamiento de tensorflow

Teniendo en cuenta lo explicado anteriormente, el uso de la librería Tensorflow fue una herramienta vital para realizar un estudio respecto a su efectividad en el reconocimiento de especies caninas y su posterior raza identificada con dicho sistema de aprendizaje.

Para dicho estudio o test de efectividad de la librería se tuvo en cuenta una muestra de 120 razas con sus respectivas imágenes teniendo como un total de 26.936 imágenes analizadas las cuales fueron analizadas mediante el diseño y despliegue de una aplicación web para facilitar el proceso de la recopilación de datos:

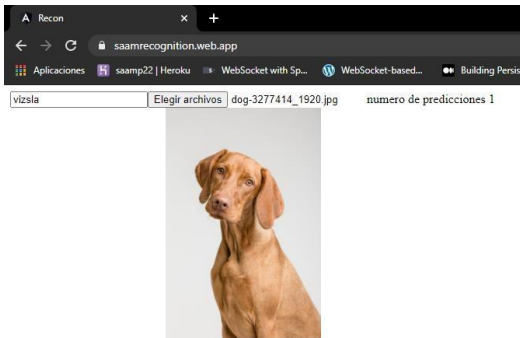
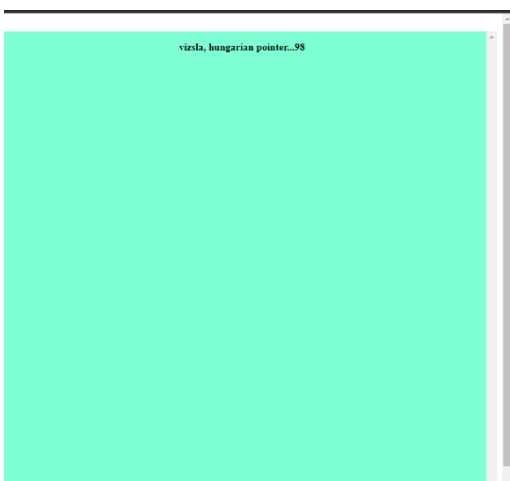


Figura 8. Ambiente de pruebas del algoritmo (propia)

Como se puede evidenciar en la figura 8, se puso la raza que se desea analizar dentro de la caja de texto para posteriormente seleccionar la imagen y de esta forma obtener su porcentaje de reconocimiento el cual se puede ver en la parte derecha.

Luego de ello, los datos obtenidos (cada celda de la tabla 1 hace referencia al porcentaje de reconocimiento por cada imagen de la raza



seleccionada), se insertando en una hoja de cálculo y en este punto se empezó a realizar el análisis del reconocimiento de razas a partir de ciertos criterios los cuales también serán tenidos en cuenta a la hora de hacer el análisis general de las razas:

98	98	95	97	99	72	65	67	74	20	99	94	0	82	15	75	9	89	98	99	99	72	96	90
57	81	89	93	94	97	99	92	97	97	88	9	78	93	99	87	56	99	99	98	100	50	85	98
0	67	84	97	92	99	97	96	99	99	98	41	90	94	0	27	61	86	98	100	82	94	96	83
97	95	75	9	94	87	91	97	96	98	99	10	20	0	83	4	89	29	98	98				
98	30	94	94	72	0	95	63	99	70	84	44												
9	95	98	12	53	99	0	52	76	7	93	94												
96	69	76	22	13	96	64	99	36	98	15	27												
91	32	87	51	32	91	99	38	0	90	97	69												
92	65	83	61	70	98	79	57	97	22	93	88												

Tabla1. Datos obtenidos insertados en una matriz en Excel, con mapa de calor (propia)

MUESTRA 188

CRITERIOS	RANGO	COLOR	TOTAL	%
BUENA	76-100		116	62%
SOBRESALIENTE	51-75		29	15%
INSUFICIENTE	26-50		16	9%
DEFICIENTE	0-25		27	14%
TOTAL			188	100%

Tabla2. Criterios del análisis de la matriz de la figura 8 (propia)

Como ejemplo, se tomaron los datos de la raza “BLENHEIM SPANIEL” figura 9 y la tabla 2, la cual se obtuvo con el procedimiento anterior y sus resultados fueron clasificados por los anteriores criterios, luego de ello se realizó un diagrama de cajas para evaluar la concentración de datos teniendo en cuenta los valores de la matriz anteriormente mostrada:

calculado con otra función llamada “PROMEDIO”, con los dos datos obtenidos anteriores se hizo cálculo de los datos representativos basado en los límites que se fijan en la siguiente tabla, para el límite mayor se hizo la suma de la desviación estándar al promedio limitando dicho valor a 100 puesto que es el máximo dato tenido en cuenta y para el límite inferior se le resta al promedio la desviación estándar para de esta forma conocer que tan alejados están los datos del promedio por lo cual tenemos como conclusión de que para la raza seleccionada los datos más representativos de la muestra se encuentran entre el 40% y el 100%.

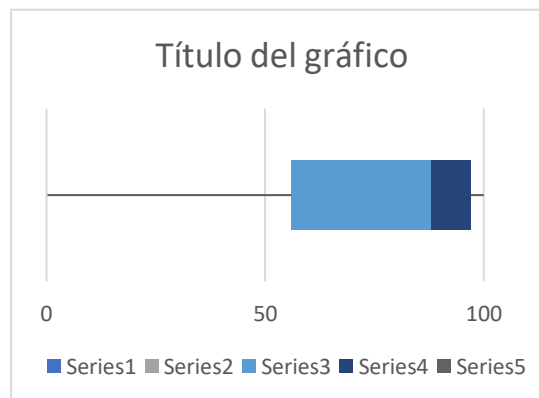


Figura 9. Diagrama de cajas de los datos de la raza “BLENHEM SPANIEL” (propia)

En este caso, se usó el diagrama anterior para poder identificar visualmente la tendencia de los datos, en este caso se concentran entre el 58% al 100% siendo los anteriores los datos más significativos, influyentes y predictivos para el reconocimiento de la misma raza basado en sus diferentes imágenes. Para la tabla 3, con la raza usada de ejemplo, se realizó un análisis dirigido específicamente a la toma de los datos significativos de los porcentajes obtenidos, donde se hace cálculo de la desviación estándar mediante una función usada de la hoja de cálculo llamada “DESVEST” y el promedio de los datos el cual fue

DESVIACIÓN ESTANDAR	
	VALOR
PROMEDIO	72
LIM. MINIMO	40
LIM. MAXIMO	100
DESVIACION EST.	32

Tabla3. Datos del diagrama de cajas de la raza “BLENHEM SPANIEL” (propia)

Con el procedimiento anterior se realizó el reconocimiento para todas las razas una por una para que de esta forma se pudiese realizar un análisis general del desempeño de la librería con todas las especies que se deseaban evaluar.

Respecto al desempeño por raza, se hizo la evaluación de su efectividad basado en 4 criterios con sus respectivos rangos los cuales serán mostrados a continuación en la tabla numero 4:

CRITERIO	MINIMO %	MAXIMO %
DEFICIENTE	0	25
INSUFICIENTE	26	50
ACEPTABLE	51	75
SOBRESALIENTE	76	100

Tabla 4. Criterios en que se evaluaron en matriz de calor (propia)

Con los datos que se obtuvieron en la forma de la tabla 4 se analizaron sus rangos de los datos teniendo en cuenta el criterio y se clasificaron de la siguiente manera:

RECONOCIMIENTO INSATISFACTORIO	RECONOCIMIENTO SATISFACTORIO
DEFICIENTE + INSUFICIENTE	ACEPTABLE + SOBRESALIENTE

Tabla 5. Como se clasificaron los datos (propia)

1. Resultados de las pruebas

Continuando con la explicación se obtuvieron los siguientes resultados:

TOTAL RAZAS ANALIZADAS	120	
TOTAL IMÁGENES	26936	
TOTAL RECONOCIMIENTO SATISFACTORIO	83	EN %
TOTAL RECONOCIMIENTO INSATISFACTORIO	37	EN %
TOTAL	120	TOTAL

Tabla 6. resultados que da el algoritmo (propia)

Ya teniendo la información anterior y teniendo en cuenta la evaluación con dichos criterios por raza como margen de error para el reconocimiento de razas.



Figura 11. Margen de error en el reconocimiento total de razas (propia)

se tuvo el siguiente resultado entre la cantidad de los reconocimientos satisfactorios e insatisfactorios para cada una de ellas, como lo podemos ver en la siguiente figura:

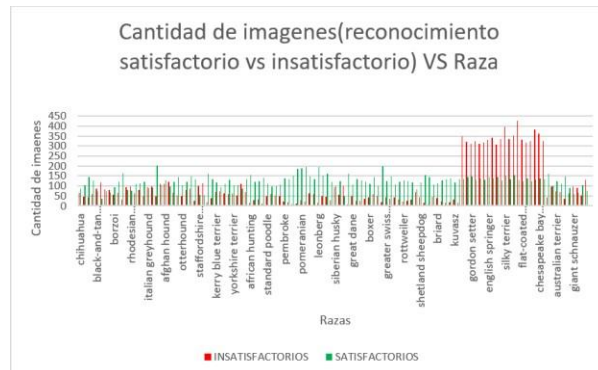


Figura 10. Resultados del algoritmo (propia)

Visto lo anterior, podemos evidenciar que para la mayoría de razas el resultado para el reconocimiento satisfactorio con la librería es superior a las cantidades de reconocimiento insatisfactorio exceptuando razas como sylky terrier o gordon setter en las cuales se evidencia un alto porcentaje inverso a lo esperado donde prima el error sobre los resultados válidos.

Se detecto de que se encontraron valores nulos en el reconocimiento, es decir, por algunos motivos se van a explicar más adelante es reconocimiento fue 0%, esto afecta los resultados positivos como se puede ver en la figura 10, por esta razón se tomaron

Como se puede ver en la figura 11, para cada raza se restaron los datos nulos (resultados iguales al 0%), a la cantidad de datos reconocidos insatisfactoriamente lo cual marca un porcentaje de dicho margen de erros para cada raza y por consiguiente se puede visualizar en la figura 12 de la siguiente forma:

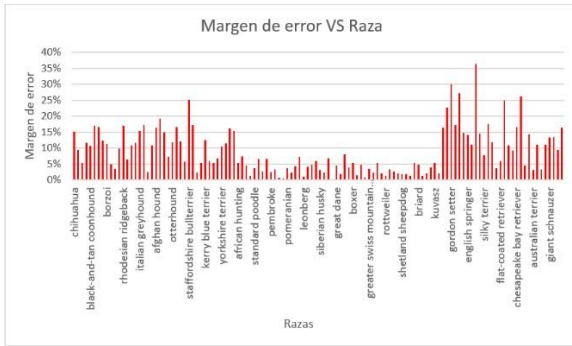


Figura 12. Datos que no obtuvieron un 0% de reconocimiento (propia)

Por el caso anterior de datos nulos en las muestras se decidió tener en cuenta únicamente los datos que tuvieron por lo menos de un 1% en adelante para el estudio con fines de ajustar los datos y evaluar su respectivo desempeño por raza de parte de la librería con el fin de reducir el margen de error mejorando el escenario de pruebas con datos que tengan un mínimo % de reconocimiento.

Con lo contado anteriormente, se hizo el respectivo ajuste a todas las razas y en esta ocasión se obtuvieron nuevos datos respecto a los valores de reconocimiento satisfactorio e insatisfactorio en



Figura13. Representación de los nuevos datos (propia)

Respecto a la cantidad de razas con un reconocimiento satisfactorio se pudo evidenciar una mejora significativa aumentando en hasta en un 9% en la figura 13 y figura 15 de efectividad el reconocimiento para el total de las 120 razas

general, lo cual los datos obtenidos ajustados se ven de la siguiente forma:

TOTAL RAZAS ANALIZADAS	120		
TOTAL IMÁGENES RECONOCIDAS	24073		
TOTAL RECONOCIMIENTO SATISFACTORIOS	94	EN %	78%
TOTAL RECONOCIMIENTO INSATISFACTORIO	26	EN %	22%
TOTAL	120	TOTAL	100%

Tabla 7. Nuevos datos (propia)

A partir de lo anterior, ahora el margen de error se toma como todos los reconocimientos que no son satisfactorios para el reconocimiento de una raza, es decir, todos los resultados inferiores por raza que sean menores a un 50% lo cual nos arroja la siguiente información:

MARGEN DE ERROR DE RECONOCIMIENTO TOTAL	%	CRITERIO
RECONOCIMIENTO INSATISFACTORIO	39%	datos < 50%
RECONOCIMIENTO SATISFACTORIO	61%	datos > 50%

Tabla 8. Nuevos datos satisfactorios e insatisfactorios (propia)

Dando como un resultado una mejora bastante significativa por raza respecto a los datos que demuestran tener un porcentaje de reconocimiento satisfactorio mucho mayor a los insatisfactorios lo cual muestra en la figura 16 que para el reconocimiento de razas la librería cumple con su función de predecir la raza en su mayoría y por ende, su trabajo es ampliamente viable para hacer uso de la misma para el reconocimiento de razas, a continuación, se mostrará gráficamente como quedarán los datos ajustados distribuidos en sus dos categorías, insatisfactorios (1% a 50%) y satisfactorios (51% a 100%):

propuestas, teniendo en cuenta lo anterior, se replantearon todos los datos de reconocimiento satisfactorio e insatisfactorio por raza, los cuales quedaron estipulados de la siguiente forma:

Reconocimientos insatisfactorios se puede evidenciar en la figura 13:



Figura 14. Nuevos Datos insatisfactorios (propia)

Reconocimientos satisfactorios:



Como se puede evidenciar, la diferencia entre los datos que están ajustados en comparación a los que no lo están en la figura 16, permiten ver que el desempeño de la librería es ampliamente viable para el reconocimiento de las 120 razas analizadas, las cuales 94 razas de las 120 serán reconocidas eficientemente por medio la librería Tensorflow y las restantes tendrán un porcentaje de reconocimiento más bajo respecto a lo esperado, por otro lado, para las razas que no fueron reconocidas se llegó a la conclusión de que hay factores que dificultan el reconocimiento de las mascotas, casos tales como características morfológicas similares entre las mismas razas y otros problemas como la falta de calidad de la imagen, resolución insuficiente en las imágenes, objetos en primer plano ajenos a las mascotas,

Figura 15. Nuevos Datos satisfactorios (propia)

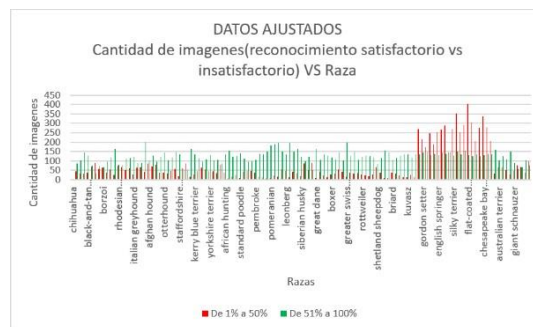


Figura 16. Datos clasificados en satisfactorios e insatisfactorios (propia)

elementos en mayor dimensión que las mascotas, fotografías movidas o desenfocadas, entre otros.

III. Conclusiones

Por consiguiente, se hizo un análisis del porqué las predicciones arrojaban un 0% o datos inferiores al 50% y que evidentemente afectan el desempeño de la librería, a continuación, se muestran los diferentes casos y su probabilidad de que sean datos insuficientes o inferiores al 50% haciendo referencia a su posible porcentaje de reconocimiento para cada problemática cuando dichas imágenes con problemas sean testeadas en el sistema de aprendizaje automático:

MOTIVOS DEL NO RECONOCIMIENTO	PROBABLES RANGOS QUE SE OBTENDRÁN
MORFOLOGÍA SIMILAR ENTRE RAZAS (características físicas similares entre las razas)	APUNTAN A VALORES ENTRE 25% Y 50%
RESOLUCION EN PX DEFICIENTE (Fotografías con muy baja resolución)	APUNTAN A VALORES ENTRE 0% Y 25%
FOTOGRAFÍAS MOVIDAS (Fotografías capturadas en movimiento)	APUNTAN A VALORES ENTRE 0% Y 25%
OBJETOS EN MAYOR DIMENSIÓN (Fotografías con objetos más significativos que el animal canino)	APUNTAN A VALORES ENTRE 25% Y 50%
FOTOGRAFÍAS DESENFOCADAS (Fotografías fuera de foco por problemas de luz u oscuridad al momento de ser capturadas)	APUNTAN A VALORES ENTRE 0% Y 25%
POSICIÓN DEFICIENTE PARA RECONOCER (Fotografías con la mascota en posiciones complicadas de realizar reconocimiento, ej: fotos de espalda, sin dirección a la cara del animal canino).	APUNTAN A VALORES ENTRE 0% Y 25%
OBJETOS DIFERENTES EN PRIMER PLANO (Fotografías con objetos en primer plano que obstruyan el reconocimiento de la mascota y al contrario reconozcan el objeto)	APUNTAN A VALORES ENTRE 25% Y 50%

Tabla 9. Análisis de porque la librería nos da 0% en el reconocimiento de razas (propia)

De acuerdo con el anterior análisis podemos dar soluciones para que el algoritmo y el sistema de aprendizaje automático tensorflow cumpla a cabalidad su funcionamiento de reconocimiento de razas con respecto a su problemática:

MOTIVOS DEL NO RECONOCIMIENTO	SOLUCIONES
COINCIDENCIA ENTRE RAZAS	AMPLIAR LOS RESULTADOS DE RECONOCIMIENTO (Devolver 3 razas en total por imagen para no discriminar predicciones basado en las similitudes morfológicas entre las razas)
RESOLUCION EN PX DEFICIENTE	SOLICITAR IMÁGENES EN BUENA RESOLUCIÓN (La aplicación contará con la disponibilidad en dispositivos Android superiores a 8.0 lo cual garantizará de las imágenes capturadas tengan buena resolución)
FOTOGRAFÍAS MOVIDAS	SOLICITAR FOTOGRAFÍAS LEGIBLES (Se dará la indicación de que las fotografías cargadas no contengan estos errores, de lo contrario al no existir reconocimiento no se subirá la mascota)
OBJETOS EN MAYOR DIMENSIÓN	SOLICITAR FOTOGRAFIA DE LA MASCOTA ÚNICAMENTE (Se dará la indicación de cómo debe ir la mascota en la fotografía para facilitar el reconocimiento)
FOTOGRAFÍAS DESENFOCADAS	SOLICITAR FOTOGRAFÍAS ENFOCADAS (Se dará la indicación de que las fotografías sean legibles a la hora del reconocimiento, de lo contrario sin reconocimiento no se subirá la mascota al sistema)
POSICIÓN DEFICIENTE PARA RECONOCER	SOLICITAR UNA POSICION ESPECIFICA PARA EL RECONOCIMIENTO (Se ha de sugerir una imagen de guía para facilitar el proceso de reconocimiento)
OBJETOS DIFERENTES EN PRIMER PLANO	SOLICITAR QUE SOLO EXISTA UN PLANO Y QUE SEA DE LA MASCOTA (Se dará la indicación de cómo debe estar la mascota en la fotografía para facilitar el proceso de reconocimiento)

Tabla 10. Soluciones para que la librería y el algoritmo hagan el reconocimiento de manera efectiva (propia)

Finalmente se plantea que la posición, definición, resolución de la imagen sea de la siguiente forma ya que es la solución a estos problemas de reconocimiento por parte del sistema:



Figura 17. Como se debe subir una imagen para que el algoritmo y la librería realicen el reconocimiento de la raza (propia)

Referencias

[1] A. Patel, L. Cheung, N. Khatod, I. Matijosaitiene, A. Arteaga, and GilkeyJoseph, “Revealing the Unknown: Real-Time Recognition of Galápagos Snake Species Using Deep Learning,” vol. 10, 2020, [Online]. Available: <https://www.mdpi.com/2076-2615/10/5/806/htm>.

[2] M.-F. Tsai, L. Pei-Ching, Z.-H. Huang, and L. Cheng-Husun, “Multiple Feature Dependency Detection for Deep Learning Technology—Smart Pet Surveillance System Implementation,” 9, 2020, [Online]. Available: <https://www.mdpi.com/2079-9292/9/9/1387/htm>.

[3] “TensorFlow, why was python the chosen language?,” 2020. https://stackoverflow.com/questions/35677724/tensorflow-why-was-python-the-chosen-language?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa.

[4] Y. H. Liu, “Feature Extraction and Image Recognition with Convolutional Neural Networks,” 2018.

- [5] J. Wu, "Introduction to Convolutional Neural Networks," 2017, [Online]. Available: <https://cs.nju.edu.cn/wujx/paper/CNN.pdf>.
- [6] M. A. López Pacheco, "Identificación de sistemas no lineales con redes neuronales convolucionales," 2017, [Online]. Available: <https://www.ctrl.cinvestav.mx/~yuw/pdf/MaTesMLP.pdf>.
- [7] D. Conde Ortiz, "Inteligencia artificial con TensorFlow para predicción de comportamientos," 2018, [Online]. Available: <https://idus.us.es/handle/11441/80122>.
- [8] A. Mejias Gil, "Herramientas modernas en redes neuronales La librería tensorflow de google," 2017, [Online]. Available: <https://repositorio.uam.es/handle/10486/679326>.
- [9] S. Minseok, J. Hyeyoom, L. Seungyoung, K. Donghyeon, and A. Minkyu, "Diagnostic Classification and Biomarker Identification of Alzheimer's Disease with Random Forest Algorithm," vol. 11, 2021, [Online]. Available: <https://www.mdpi.com/2076-3425/11/4/453>.
- [10] M. Syed, D. Kaushik, D. Pranab Kumar, and Takeshi Koshiba, "Plant Leaf Disease Recognition Using Depth-Wise Separable Convolution-Based Models," vol. 13, 2021, [Online]. Available: <https://www.mdpi.com/2073-8994/13/3/511>.
- [11] Stanford University, "Stanford Dogs Dataset," [Online]. Available: <http://vision.stanford.edu/aditya86/ImageNetDogs/>.
- [12] A. Khosla, N. Jayadevaprakash, Y. Bangpeng, and F.-F. Li, "Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs," [Online]. Available: <http://vision.stanford.edu/aditya86/ImageNetDogs/>.
- [13] I. Lorencin *et al.*, "On Urinary Bladder Cancer Diagnosis: Utilization of Deep Convolutional Generative Adversarial Networks for Data Augmentation," 2021, [Online]. Available: <https://www.mdpi.com/2079-7737/10/3/175>.
- [14] M. Massiris, C. Delrieux, and J. Á. Fernández Muñoz, "Detección de equipos de protección personal mediante red neuronal convolucional YOLO," 2018, [Online]. Available: <https://ruc.udc.es/dspace/handle/2183/24891>.
- [15] M. Pacheco, Y. Wen, and J. Morales, "Detección de daño en edificios basada en datos de aceleración y redes neuronales convolucionales," [Online]. Available: <http://www.amca.mx/RevistaDigital/cnca2019/files/0075.pdf>.
- [16] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," p. 10, 2015, [Online]. Available: <https://arxiv.org/abs/1511.08458>.
- [17] M. Abadí, M. Isard, and D. Murray, "A computational model for TensorFlow: an introduction," 2017, [Online]. Available: <https://dl.acm.org/doi/abs/10.1145/3088525.3088527>.
- [18] Y. H. Liu, "Feature Extraction and Image Recognition with Convolutional Neural Networks," 2018.
- [19] R. Feng, Z. Xing, Z. Dawei, X. Zhanyang, W. Shaohua, and Q. Lianyong, "Deep learning for real-time image steganalysis: a survey," 2019, [Online]. Available: <https://link.springer.com/article/10.1007/s11554-019-00915-5>.
- [20] R. Girshick, D. Jeff, D. Trevor, and M. Jitendra, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," vol. 38, [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7112511>.
- [21] R. B. Z. R. O'REILLY, "TensorFlow for Deep learning," 2018.

